The Second Report of PRML

Liu Jitao
22371262@buaa.edu.cn

In this report, I use 3 methods to classify a 3D data set which content 1000 points, and test these methods with a identically distributed data set and discuss their performance. All the codes are published on `https://github.com/KevinTJL/PRML-2025/tree/main/assignment_2`.

I. Trianing Data

There are 1000 3D points in the training data set. They are shown in graph below.

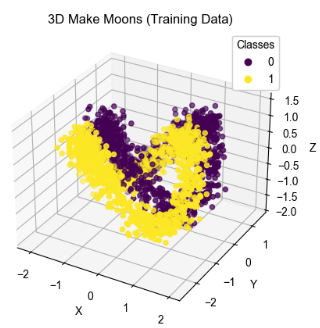

Fig. 1 This picture shows training data set

II. Decision Tree

A. introduction

A decision tree is a supervised machine learning algorithm used for both classification and regression tasks. It works by splitting the dataset into smaller subsets based on feature values, creating a tree-like structure of decisions. The following equation gives the mathematical model of Decision trees.

$$Gini(D) = 1 - \sum_{k=1}^{K} p_k^2 \tag{1}$$

$$\Delta Gini = Gini(D) - \sum_{i=1}^{m} \frac{|D_i|}{|D|} Gini(D_i) \tag{2}$$

As equation shows, our method is to maximize the $\Delta Gini$ to achieve the classify goal.

B. Results

In this section I use decision trees as the classify method. By training with given data and test with the identically distributed data, the accuracy of Decision trees reach 94% and the classification result is shown in the graph bellow.

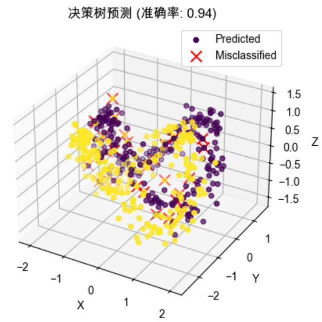

Fig. 2 This picture shows the result of Decision Trees

III. Adaboost+Decision Tree

A. Introduction

AdaBoost (Adaptive Boosting) is an ensemble learning technique that combines multiple weak learners (typically decision trees) to create a strong classifier. It works by iteratively training models, focusing more on misclassified samples in each round, and then combining their predictions through weighted voting.

$$\epsilon_t = \sum_{i=1}^{N} w_i \cdot \mathbb{I}(y_i \neq \hat{y}_i) \tag{3}$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{4}$$

$$w_i \leftarrow w_i \cdot e^{\alpha_t \cdot \mathbb{I}(y_i \neq \hat{y}_i)} \tag{5}$$

$$F(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right) \tag{6}$$

By the formulas above, we can inform the prediction of Tree-t as $F(x)$.

B. Results

The precession of Adaboost+Decision tree method is 72%. Figure is shown below.

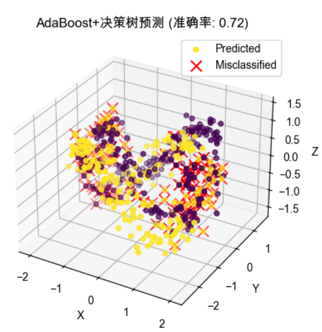

Fig. 3 This picture shows the result of Decision Trees with adaboost

IV. SVM

A. Introduction

Support Vector Machines (SVM) is a powerful supervised learning algorithm used for classification and regression. Its key idea is to find the optimal hyperplane that best separates data points of different classes while maximizing the margin (distance to the nearest data points).

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{7}$$

Where C is a regularization parameter controlling trade-off

Maps data into a higher-dimensional space need to use a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. And there are 3 different kernel tricks in the following subsections

B. Results with linear kernel

With linear kernel shown in (8) we can get a method with 68% accuracy.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j \tag{8}$$

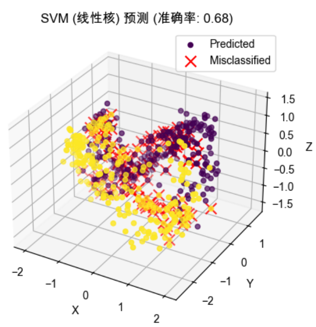

Fig. 4 This picture shows the result of SVM with linear kernel

C. Results with polynomial kernel

With linear kernel shown in (9) we can get a method with 87% accuracy.

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma\mathbf{x}_i^T\mathbf{x}_j + r)^d \tag{9}$$

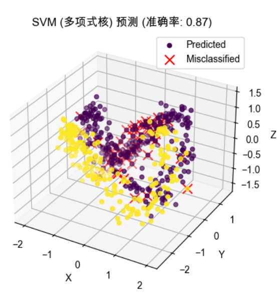

Fig. 5 This picture shows the result of SVM with polynomial kernel

D. Results with RBF kernel

With linear kernel shown in (10) we can get a method with 98% accuracy.

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \tag{10}$$

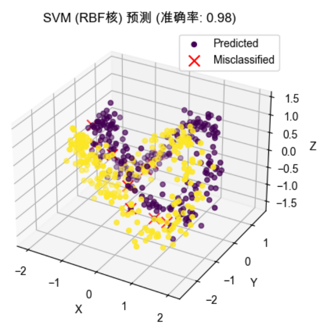

Fig. 6 This picture shows the result of SVM with RBF kernel

V. Discussion and Conclusion

A. Discussion

The results are very impressive, in common sense, adaboost+decision trees as the advance version of decision tree which should achieve a better performance than the early version, but it fails. It may because the noise of both training data and test data are too weak, So I make an extra experiment to check which affect the performance.

	Decision Tree	Adaboost + Decision Trees	SVM Linear Kernel	SVM Polynomial Kernel	SVM RBF Kernel
Train noise: 0.2 Test noise: 0.2	0.94	0.72	0.68	0.87	0.98
Train noise: 1 Test noise: 0.2	0.55	0.67	0.68	0.68	0.67
Train noise: 0.2 Test noise: 1	0.62	0.61	0.64	0.64	0.63
Train noise: 2 Test noise: 2	0.51	0.57	0.59	0.58	0.59

Table 1 This table shows the accuracy score of different methods in diverse noises.

We can learn that the accuracy of Decision Tree drops sharply when noise increases, but the other methods remain almost unchanged. It may because the tree structure leads to an incline to overfit when the training data contains high noise, but performance well in clean data. The other algorithm introduce adaptive or other methods to reduce the effect of noises, but sacrifice some behavior.

B. Concolusion

In this 3D classification task, Decision Trees excel with low noise (0.94 accuracy) but degrade sharply under noise. AdaBoost + Trees shows moderate robustness (0.67→0.57). SVMs, especially RBF kernel (0.98 clean, 0.59 noisy), deliver consistent performance across noise levels, proving most reliable. Key takeaway: Use Decision Trees for clean data, SVMs for noisy or uncertain environments, and AdaBoost as a balanced compromise.